



PRELIMINARY

DATA

SHEET

MARCH, 1976

### MCS6530 (MEMORY, I/O, TIMER ARRAY)

The MCS6530 is designed to operate in conjunction with the MCS650X Microprocessor Family. It is comprised of a mask programmable 1024 x 8 ROM, a 64 x 8 static RAM, two software controlled 8 bit bi-directional data ports allowing direct interfacing between the microprocessor unit and peripheral devices, and a software programmable interval timer with interrupt, capable of timing in various intervals from 1 to 262,144 clock periods.

- \* 8 bit bi-directional Data Bus for direct communication with the microprocessor
- \* 1024 x 8 ROM
- \* 64 x 8 static RAM
- \* Two 8 bit bi-directional data ports for interface to peripherals
- \* Two programmable I/O Peripheral Data Direction Registers
- \* Programmable Interval Timer
- \* Programmable Interval Timer Interrupt
- \* TTL & CMOS compatible peripheral lines
- \* Peripheral pins with Direct Transistor Drive Capability
- \* High Impedance Three-State Data Pins
- \* Allows up to 7K contiguous bytes of ROM with no external decoding

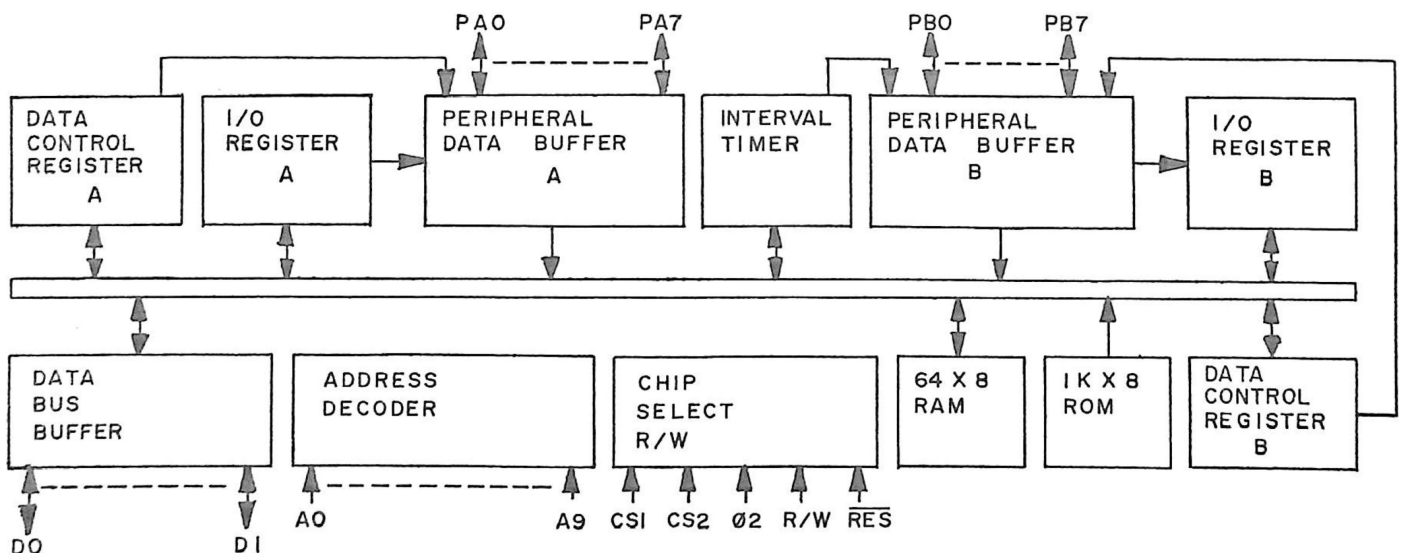


Figure 1. MCS6530 Block Diagram

## MAXIMUM RATINGS

RATING	SYMBOL	VOLTAGE	UNIT
Supply Voltage	VCC	-.3 to +7.0	V
Input/Output Voltage	V <sub>IN</sub>	-.3 to +7.0	V
Operating Temperature Range	T <sub>OP</sub>	0 to 70	°C
Storage Temperature Range	T <sub>STG</sub>	-55 to +150	°C

All inputs contain protection circuitry to prevent damage due to high static charges. Care should be exercised to prevent unnecessary application of voltage outside the specification range.

## ELECTRICAL CHARACTERISTICS (VCC = 5.0v ± 5%, VSS = 0v, T<sub>A</sub> = 25° C)

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT
Input High Voltage	V <sub>IH</sub>	V <sub>SS</sub> +2.4		VCC	V
Input Low Voltage	V <sub>IL</sub>	V <sub>SS</sub> -.3		V <sub>SS</sub> +.4	V
Input Leakage Current; V <sub>IN</sub> = V <sub>SS</sub> + 5v A0-A9, RS, R/W, RES, 02, PB6*, PB5*	I <sub>IN</sub>		1.0	2.5	µA
Input Leakage Current for High Impedance State (Three State); V <sub>IN</sub> = .4v to 2.4v; D0-D7	I <sub>TSI</sub>		±1.0	±10.0	µA
Input High Current; V <sub>IN</sub> = 2.4v PA0-PA7, PB0-PB7	I <sub>IH</sub>	-100.	-300.		µA
Input Low Current; V <sub>IN</sub> = .4v PA0-PA7, PB0-PB7	I <sub>IL</sub>		-1.0	-1.6	MA
Output High Voltage VCC = MIN, I <sub>LOAD</sub> ≤ -100µA (PA0-PA7, PB0-PB7, D0-D7) I <sub>LOAD</sub> ≤ -3 MA (PA0, PB0)	V <sub>OH</sub>			V <sub>SS</sub> +2.4 V <sub>SS</sub> +1.5	V
Output Low Voltage VCC = MIN, I <sub>LOAD</sub> ≤ 1.6MA	V <sub>OL</sub>			V <sub>SS</sub> +.4	V
Output High Current (Sourcing); V <sub>OH</sub> ≥ 2.4v (PA0-PA7, PB0-PB7, D0-D7) ≥ 1.5v Available for other than TTL (Darlington) (PA0, PB0)	I <sub>OH</sub>	-100 -3.0	-1000 -5.0		µA MA
Output Low Current (Sinking); V <sub>OL</sub> ≤ .4v (PA0-PA7) (PB0-PB7)	I <sub>OL</sub>	1.6			MA
Clock Input Capacitance	C <sub>Clk</sub>			30	pf
Input Capacitance	C <sub>IN</sub>			10	pf
Output Capacitance	C <sub>OUT</sub>			10	pf
Power Dissipation	P <sub>D</sub>		500	1000	MW

\*When programmed as address pins  
All values are D.C. readings

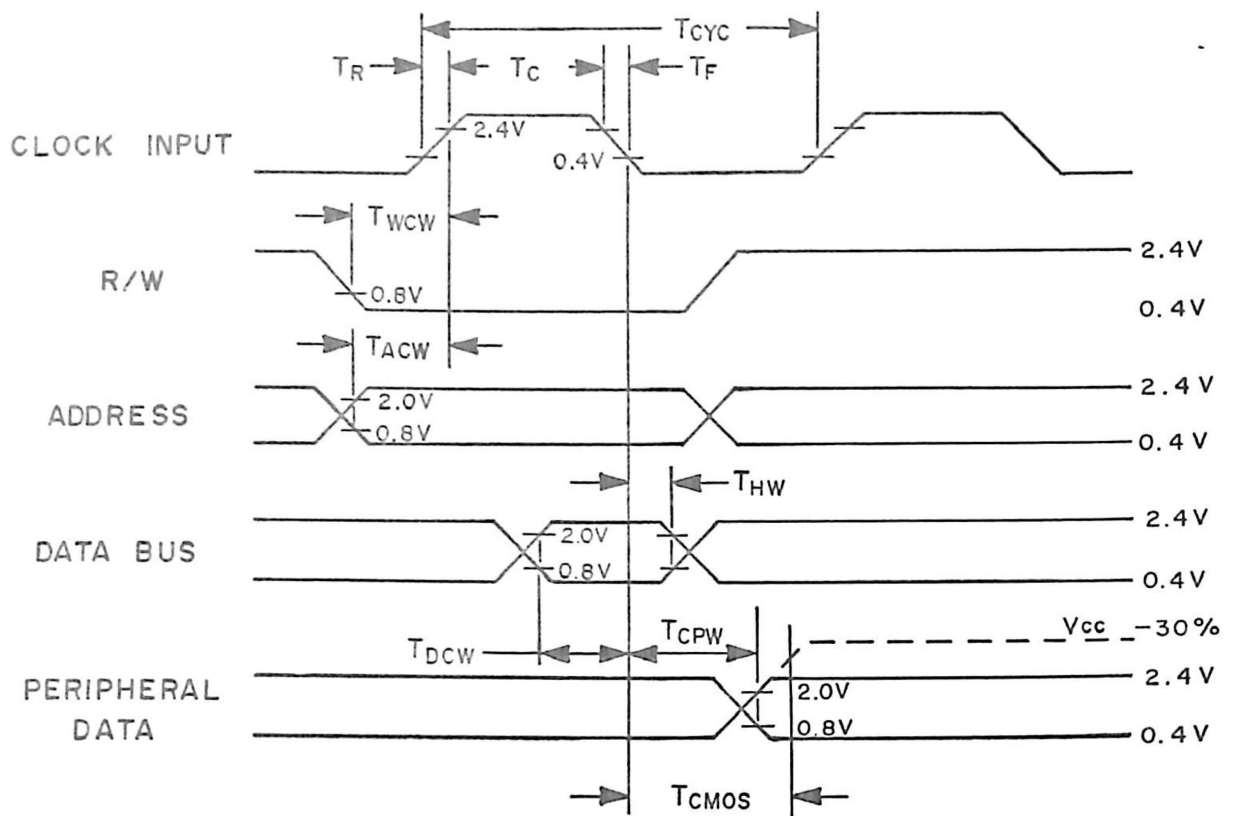
### WRITE TIMING CHARACTERISTICS

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT
Clock Period	$T_{CYC}$	1		10	$\mu S$
Rise & Fall Times	$T_R, T_F$			25	NS
Clock Pulse Width	$T_C$	470			NS
R/W valid before positive transition of clock	$T_{WCW}$	180			NS
Address valid before positive transition of clock	$T_{ACW}$	180			NS
Data Bus valid before negative transition of clock	$T_{DCW}$	300			NS
Data Bus Hold Time	$T_{HW}$	10			NS
Peripheral data valid after negative transition of clock	$T_{CPW}$			1	$\mu S$
Peripheral data valid after negative transition of clock driving CMOS (Level= $V_{CC}-30\%$ )	$T_{CMOS}$			2	$\mu S$

### READ TIMING CHARACTERISTICS

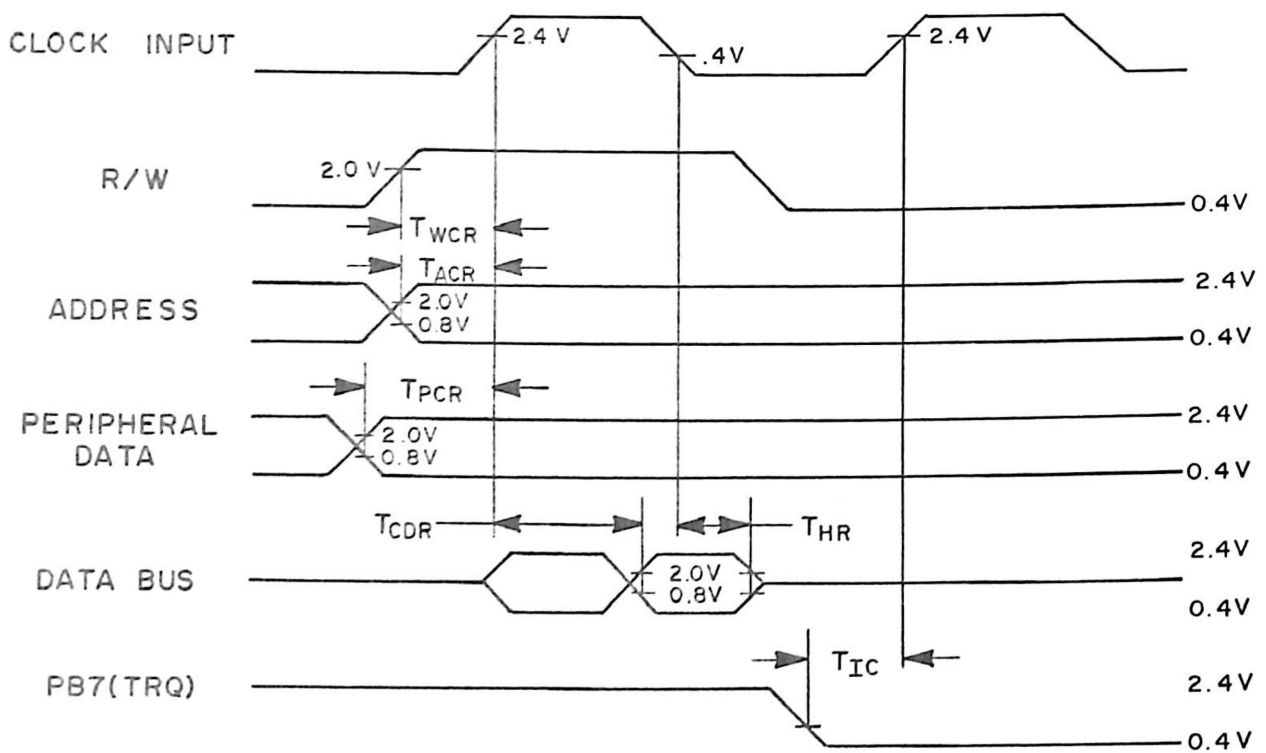
CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT
R/W valid before positive transition of clock	$T_{WCR}$	180			NS
Address valid before positive transition of clock	$T_{ACR}$	180			NS
Peripheral data valid before positive transition of clock	$T_{PCR}$	300			NS
Data Bus valid after positive transition of clock	$T_{CDR}$			395	NS
Data Bus Hold Time	$T_{HR}$	10			NS
$\overline{IRQ}$ (Interval Timer Interrupt) valid before positive transition of clock	$T_{IC}$	200			NS

Loading = 30 pf + 1 TTL load for PA $\emptyset$ -PA7, PB $\emptyset$ -PB7  
 =130 pf + 1 TTL load for D $\emptyset$ -D7



WRITE TIMING CHARACTERISTICS

Figure 2



READ TIMING CHARACTERISTICS

Figure 3

## INTERFACE SIGNAL DESCRIPTION

### Reset (RES)

During system initialization a Logic "0" on the  $\overline{\text{RES}}$  input will cause a zeroing of all four I/O registers. This in turn will cause all I/O buses to act as inputs thus protecting external components from possible damage and erroneous data while the system is being configured under software control. The Data Bus Buffers are put into an OFF-STATE during Reset. Interrupt capability is disabled with the RES signal. The RES signal must be held low for at least one clock period when reset is required.

### Input Clock

The input clock is a system Phase Two clock which can be either a low level clock ( $V_{IL} < 0.4$ ,  $V_{IH} > 2.4$ ) or high level clock ( $V_{IL} < 0.2$ ,  $V_{IH} = V_{CC} + 0.3$ ).

### Read/Write (R/W)

The R/W signal is supplied by the microprocessor array and is used to control the transfer of data to and from the microprocessor array and the MCS6530. A high on the R/W pin allows the processor to read (with proper addressing) the data supplied by the MCS6530. A low on the R/W pin allows a write (with proper addressing) to the MCS6530.

### Interrupt Request (IRQ)

The  $\overline{\text{IRQ}}$  pin is an interrupt pin from the interval timer. This same pin, if not used as an interrupt, can be used as a peripheral I/O pin (PB7). When used as an interrupt, the pin should be set up as an input by the data direction register. The pin will be normally high with a low indicating an interrupt from the MCS6530. An external pull-up device is not required; however, if collector-OR'd with other devices, the internal pullup may be omitted with a mask option.

### Data Bus (D0-D7)

The MCS6530 has eight bi-directional data pins (D0-D7). These pins connect to the system's data lines and allow transfer of data to and from the microprocessor array. The output buffers remain in the off state except when a Read operation occurs.

### Peripheral Data Ports

The MCS6530 has 16 pins available for peripheral I/O operations. Each pin is individually software programmable to act as either an input or an output. The 16 pins are divided into 2 8-bit ports, PA0-PA7 and PB0-PB7. PB5, PB6 and PB7 also have other uses which are discussed in later sections. The pins are set up as an input by writing a "0" into the corresponding bit of the data direction register. A "1" into the data direction register will cause its corresponding bit to be an output. When in the input mode, the

peripheral output buffers are in the "1" state and a pull-up device acts as less than one TTL load to the peripheral data lines. On a Read operation, the microprocessor unit reads the peripheral pin. When the peripheral device gets information from the MCS6530 it receives data stored in the data register. The microprocessor will read correct information if the peripheral lines are greater than 2.0 volts for a "1" and less than 0.8 volts for a "0" as the peripheral pins are all TTL compatible. Pins PA0 and PB0 are also capable of sourcing 3 ma at 1.5v, thus making them capable of Darlington drive.

#### Address Lines (A0-A9)

There are 10 address pins. In addition to these 10, there is the ROM SELECT pin. The above pins, A0-A9 and ROM SELECT, are always used as addressing pins. There are 2 additional pins which are mask programmable and can be used either individually or together as CHIP SELECTS. They are pins PB5 and PB6. When used as peripheral data pins they cannot be used as chip selects.

### INTERNAL ORGANIZATION

A block diagram of the internal architecture is shown in Figure 1. The MCS6530 is divided into four basic sections, RAM, ROM, I/O and TIMER. The RAM and ROM interface directly with the microprocessor through the system data bus and address lines. The I/O section consists of 2 8-bit halves. Each half contains a Data Direction Register (DDR) and an I/O Register.

#### ROM 1K Byte (8K Bits)

The 8K ROM is in a 1024 x 8 configuration. Address lines A0-A9, as well as RS0 are needed to address the entire ROM. With the addition of CS1 and CS2, seven MCS6530's may be addressed, giving 7168 x 8 bits of contiguous ROM.

#### RAM - 64 Bytes (512 Bits)

A 64 x 8 static RAM is contained on the MCS6530. It is addressed by A0-A5 (Byte Select), RS0, A6, A7, A8, A9 and, depending on the number of chips in the system, CS1 and CS2.

#### Internal Peripheral Registers

There are four internal registers, two data direction registers and two peripheral I/O data registers. The two data direction registers (A side and B side) control the direction of the data into and out of the peripheral pins. A "1" written into the Data Direction Register sets up the corresponding peripheral buffer pin as an output. Therefore, anything then written into the I/O Register will appear on that corresponding peripheral

pin. A "0" written into the DDR inhibits the output buffer from transmitting data to or from the I/O Register. For example, a "1" loaded into data direction register A, position 3, sets up peripheral pin PA3 as an output. If a "0" had been loaded, PA3 would be configured as an input and remain in the high state. The two data I/O registers are used to latch data from the Data Bus during a Write operation until the peripheral device can read the data supplied by the microprocessor array.

During a read operation the microprocessor is not reading the I/O Registers but in fact is reading the peripheral data pins. For the peripheral data pins which are programmed as outputs the microprocessor will read the corresponding data bits of the I/O Register. The only way the I/O Register data can be changed is by a microprocessor Write operation. The I/O Register is not affected by a Read of the data on the peripheral pins.

### Interval Timer

The Timer section of the MCS6530 contains three basic parts: preliminary divide down register, programmable 8-bit register and interrupt logic. These are illustrated in Figure 4.

The interval timer can be programmed to count up to 256 time intervals. Each time interval can be either 1T, 8T, 64T or 1024T increments, where T is the system clock period. When a full count is reached, an interrupt flag is set to a logic "1." After the interrupt flag is set the internal clock begins counting down to a maximum of -255T. Thus, after the interrupt flag is set, a Read of the timer will tell how long since the flag was set up to a maximum of 255T.

The 8 bit system Data Bus is used to transfer data to and from the Interval Timer. If a count of 52 time intervals were to be counted, the pattern 0 0 1 1 0 1 0 0 would be put on the Data Bus and written into the Interval Time register.

At the same time that data is being written to the Interval Timer, the counting intervals of 1, 8, 64, 1024T are decoded from address lines A0 and A1. During a Read or Write operation address line A3 controls the interrupt capability of PB7, i.e.,  $A_3 = 1$  enables IRQ on PB7,  $A_3 = 0$  disables IRQ on PB7. When PB7 is to be used as an interrupt flag with the interval timer it should be programmed as an input. If PB7 is enabled by A3 and an interrupt occurs PB7 will go low. When the timer is read prior to the interrupt flag being set, the number of time intervals remaining will be read, i.e., 51, 50, 49, etc.

When the timer has counted down to 0 0 0 0 0 0 0 0 on the next count time an interrupt will occur and the counter will read 1 1 1 1 1 1 1 1. After interrupt, the timer register decrements at a divide by "1" rate of the system clock. If after interrupt, the timer is read and a value of 1 1 1 0 0 1 0 0 is read, the time since interrupt is 28T. The value read is in two's complement.

```

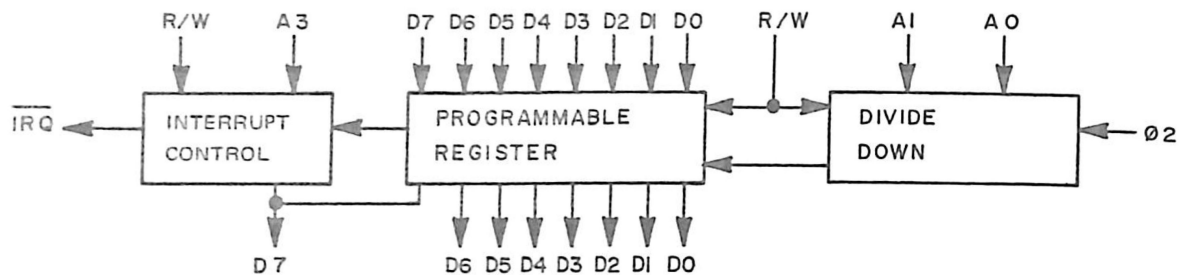
Value read = 1 1 1 0 0 1 0 0
Complement = 0 0 0 1 1 0 1 1
ADD 1      = 0 0 0 1 1 1 0 0 = 28.

```

Thus, to arrive at the total elapsed time, merely do a two's complement add to the original time written into the timer. Again, assume time written as 0 0 1 1 0 1 0 0 (=52). With a divide by 8, total time to interrupt is  $(52 \times 8) + 1 = 417T$ . Total elapsed time would be  $416T + 28T = 444T$ , assuming the value read after interrupt was 1 1 1 0 0 1 0 0.

After the interrupt, whenever the timer is written or read the interrupt is reset. However, the reading of the timer at the same time the interrupt occurs will not reset the interrupt flag. When the interrupt flag is read on DB7 all other DB outputs (DB0 thru DB6) go to "0".

Figure 5 illustrates an example of interrupt.



BASIC ELEMENTS OF INTERVAL TIMER - Figure 4

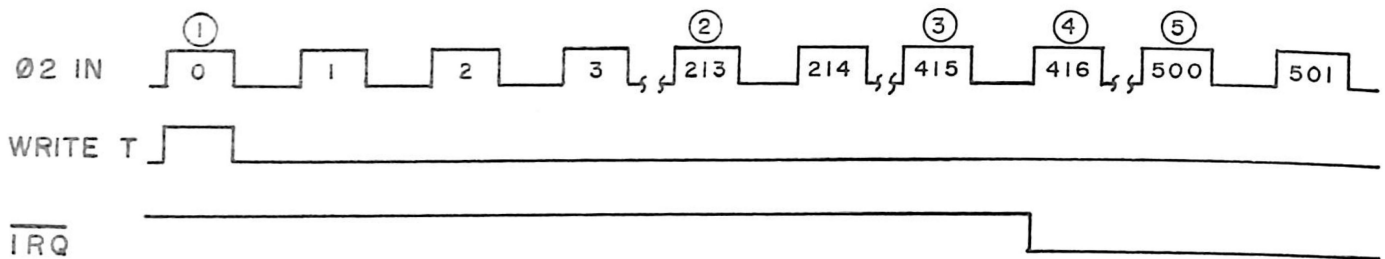


Figure 5

1. Data written into interval timer is  $00110100 = 52_{10}$
2. Data in Interval timer is  $00011001 = 25_{10}$   
 $52 - \frac{213}{8} - 1 = 52 - 26 - 1 = 25$
3. Data in Interval timer is  $00000000 = 0_{10}$   
 $52 - \frac{415}{8} - 1 = 52 - 51 - 1 = 0$
4. Interrupt has occurred at  $\emptyset_2$  pulse #416  
 Data in Interval timer = 1 1 1 1 1 1 1 1
5. Data in Interval timer is 1 0 1 0 1 1 0 0  
 two's complement is  $01010100 = 84_{10}$   
 $84 + (52 \times 8) = 500_{10}$

When reading the timer after an interrupt, A3 should be low so as to disable the IRQ pin. This is done so as to avoid future interrupts until after another Write timer operation.



## ADDRESSING

Addressing of the MCS6530 offers many variations to the user for greater flexibility. The user may configure his system with RAM in lower memory, ROM in higher memory, and I/O registers with interval timers between the extremes. There are 10 address lines (A0-A9). In addition, there is the possibility of 3 additional address lines to be used as chip-selects and to distinguish between ROM, RAM, I/O and interval timer. Two of the additional lines are chip-selects 1 and 2 (CS1 and CS2). The chip-select pins can also be PB5 and PB6. Whether the pins are used as chip-selects or peripheral I/O pins is a mask option and must be specified when ordering the part. Both pins act independently of each other in that either or both pins may be designated as a chip-select. The third additional address line is RSO. The MCS6502 and MCS6530 in a 2-chip system would use RSO to distinguish between ROM and non-ROM sections of the MCS6530. With the addressing pins available, a total of 7K contiguous ROM may be addressed with no external decode. Below is an example of a 1-chip and a 7-chip MCS6530 Addressing Scheme.

### One-Chip Addressing

Figure 6 illustrates a 1-chip system decode for the MCS6530.

### Seven-Chip Addressing

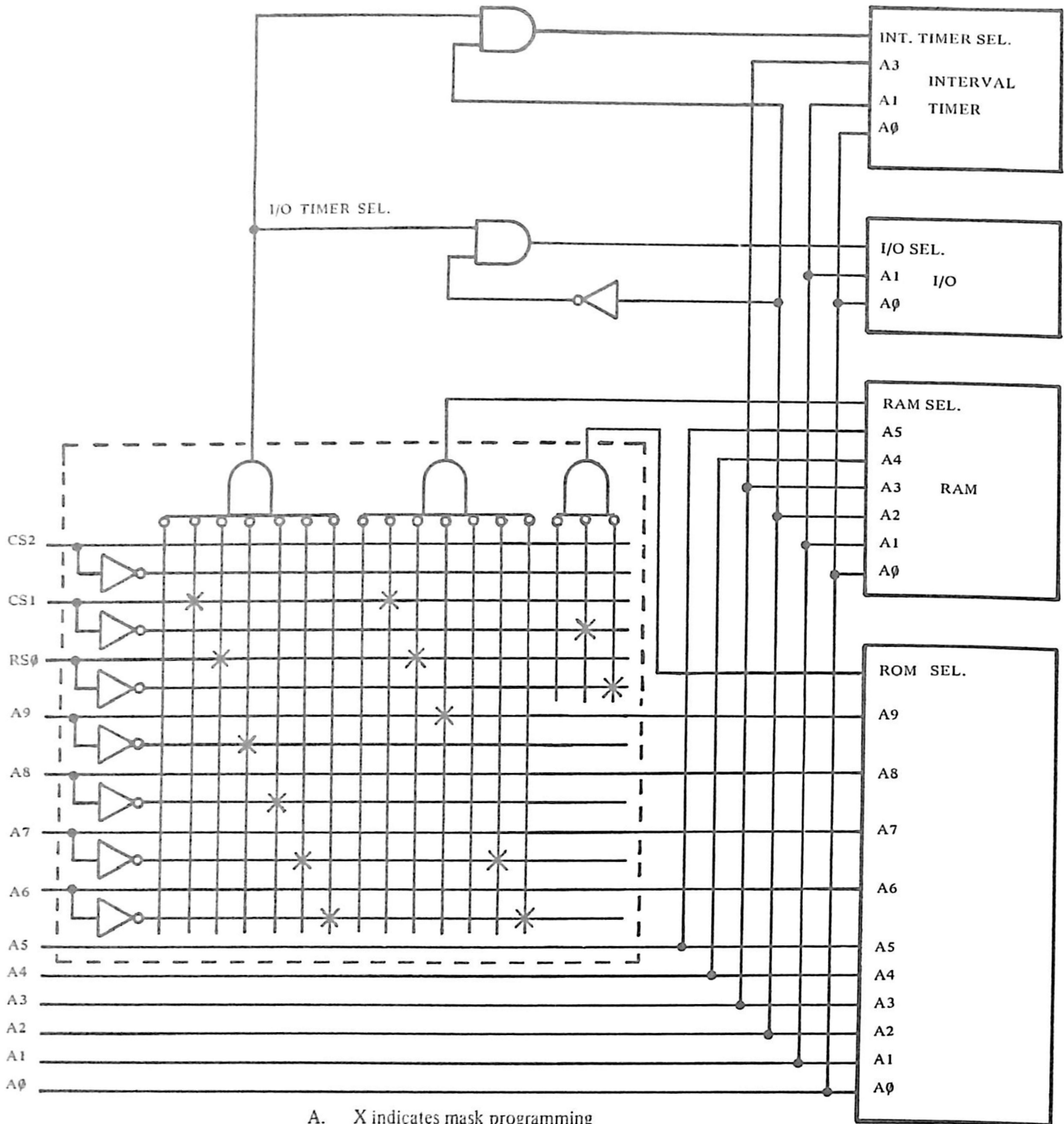
In the 7-chip system the objective would be to have 7K of contiguous ROM, with RAM in low order memory. The 7K of ROM could be placed between addresses 65,535 and 1024. For this case, assume A13, A14 and A15 are all 1 when addressing ROM, and 0 when addressing RAM or I/O. This would place the 7K ROM between Addresses 65,535 and 58,367. The 2 pins designated as chip-select or I/O would be masked programmed as chip-select pins. Pin RSO would be connected to address line A10. Pins CS1 and CS2 would be connected to address lines A11 and A12 respectively. See Figure 7.

The two examples shown would allow addressing of the ROM and RAM; however, once the I/O or timer has been addressed, further decoding is necessary to select which of the I/O registers are desired, as well as the coding of the interval timer.

### I/O Register - Timer Addressing

Figure 8 illustrates the address decoding for the internal elements and timer programming. Address lines A2 distinguishes I/O registers from the timer. When A2 is high and I/O timer select is high, the I/O registers are addressed. Once the I/O registers are addressed, address lines A1 and A0 decode the desired register.

When the timer is selected A1 and A0 decode the divide by matrix. This decoding is defined in Figure 8. In addition, Address A3 is used to enable the interrupt flag to PB7.



- A. X indicates mask programming  
 i.e. ROM select =  $CS1 \cdot RSO$   
 RAM select =  $CS1 \cdot \overline{RSO} \cdot A9 \cdot A7 \cdot A6$   
 I/O TIMER SELECT =  $CS1 \cdot \overline{RSO} \cdot A9 \cdot A8 \cdot A7 \cdot A6$
- B. Notice that A8 is a don't care for  
 RAM select
- C. CS2 can be used as PB5 in this example.

MCS6530 One Chip Address Encoding Diagram  
 Figure 6

The addressing of the ROM select, RAM select and I/O Timer select lines would be as follows:

		<u>CS2</u>	<u>CS1</u>	<u>RS<math>\emptyset</math></u>	<u>A9</u>	<u>A8</u>	<u>A7</u>	<u>A6</u>
		<u>A12</u>	<u>A11</u>	<u>A10</u>				
MCS6530 #1,	ROM SELECT	0	0	1	X	X	X	X
	RAM SELECT	0	0	0	0	0	0	0
	I/O TIMER	0	0	0	1	0	0	0
MCS6530 #2,	ROM SELECT	0	1	0	X	X	X	X
	RAM SELECT	0	0	0	0	0	0	1
	I/O TIMER	0	0	0	1	0	0	1
MCS6530 #3,	ROM SELECT	0	1	1	X	X	X	X
	RAM SELECT	0	0	0	0	0	1	0
	I/O TIMER	0	0	0	1	0	1	0
MCS6530 #4,	ROM SELECT	1	0	0	X	X	X	X
	RAM SELECT	0	0	0	0	0	1	1
	I/O TIMER	0	0	0	1	0	1	1
MCS6530 #5,	ROM SELECT	1	0	1	X	X	X	X
	RAM SELECT	0	0	0	0	1	0	0
	I/O TIMER	0	0	0	1	1	0	0
MCS6530 #6,	ROM SELECT	1	1	0	X	X	X	X
	RAM SELECT	0	0	0	0	1	0	1
	I/O TIMER	0	0	0	1	1	0	1
MCS6530 #7,	ROM SELECT	1	1	1	X	X	X	X
	RAM SELECT	0	0	0	0	1	1	0
	I/O TIMER	0	0	0	1	1	1	0

\* RAM select for MCS6530 #5 would read =  $\overline{A12} \cdot \overline{A11} \cdot \overline{A10} \cdot \overline{A9} \cdot A8 \cdot \overline{A7} \cdot \overline{A6}$

### MCS6530 Seven Chip Addressing Scheme

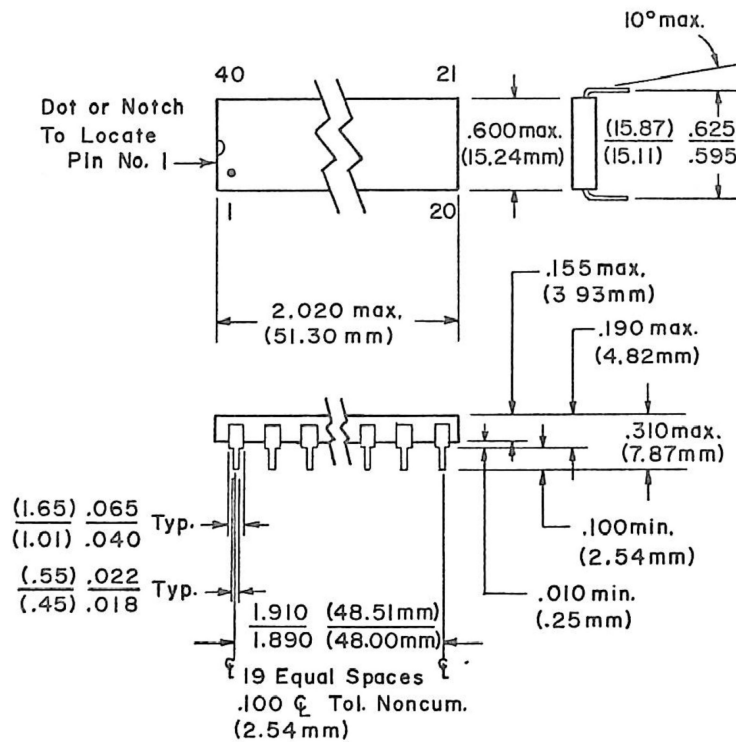
Figure 7

ADDRESSING DECODE

	<u>ROM SELECT</u>	<u>RAM SELECT</u>	<u>I/O TIMER SELECT</u>	<u>R/W</u>	<u>A3</u>	<u>A2</u>	<u>A1</u>	<u>A0</u>
READ ROM	1	0	0	1	X	X	X	X
WRITE RAM	0	1	0	0	X	X	X	X
READ RAM	0	1	0	1	X	X	X	X
WRITE DDRA	0	0	1	0	X	0	0	1
READ DDRA	0	0	1	1	X	0	0	1
WRITE DDRB	0	0	1	0	X	0	1	1
READ DDRB	0	0	1	1	X	0	1	1
WRITE PER. REG. A	0	0	1	0	X	0	0	0
READ PER. REG. A	0	0	1	1	X	0	0	0
WRITE PER. REG. B	0	0	1	0	X	0	1	0
READ PER. REG. B	0	0	1	1	X	0	1	0
WRITE TIMER								
÷ IT	0	0	1	0	*	1	0	0
÷ 8T	0	0	1	0	*	1	0	1
÷ 64T	0	0	1	0	*	1	1	0
÷ 1024T	0	0	1	0	*	1	1	1
READ TIMER	0	0	1	1	*	1	X	0
READ INTERRUPT FLAG	0	0	1	1	X	1	X	1

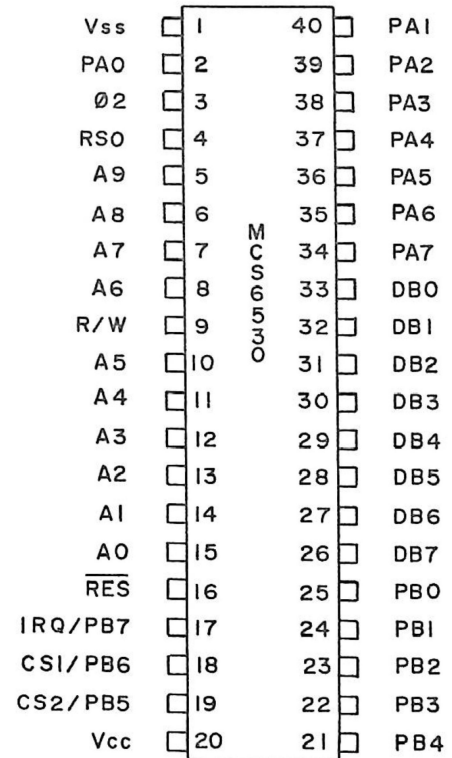
\* A<sub>3</sub> = 1 Enables IRQ to PB7  
 A<sub>3</sub> = 0 Disables IRQ to PB7

Addressing Decode for I/O Register and Timer  
 FIGURE 8



NOTE: Pin No.1 is in lower left corner when symbolization is in normal orientation

PACKAGE OUTLINE



PIN DESIGNATION

**brought to you by  
andy finkel**